

# Command Processor CSCI

## Requirements Review

May 8,1997

Version 1.1

# Software Requirements Specification

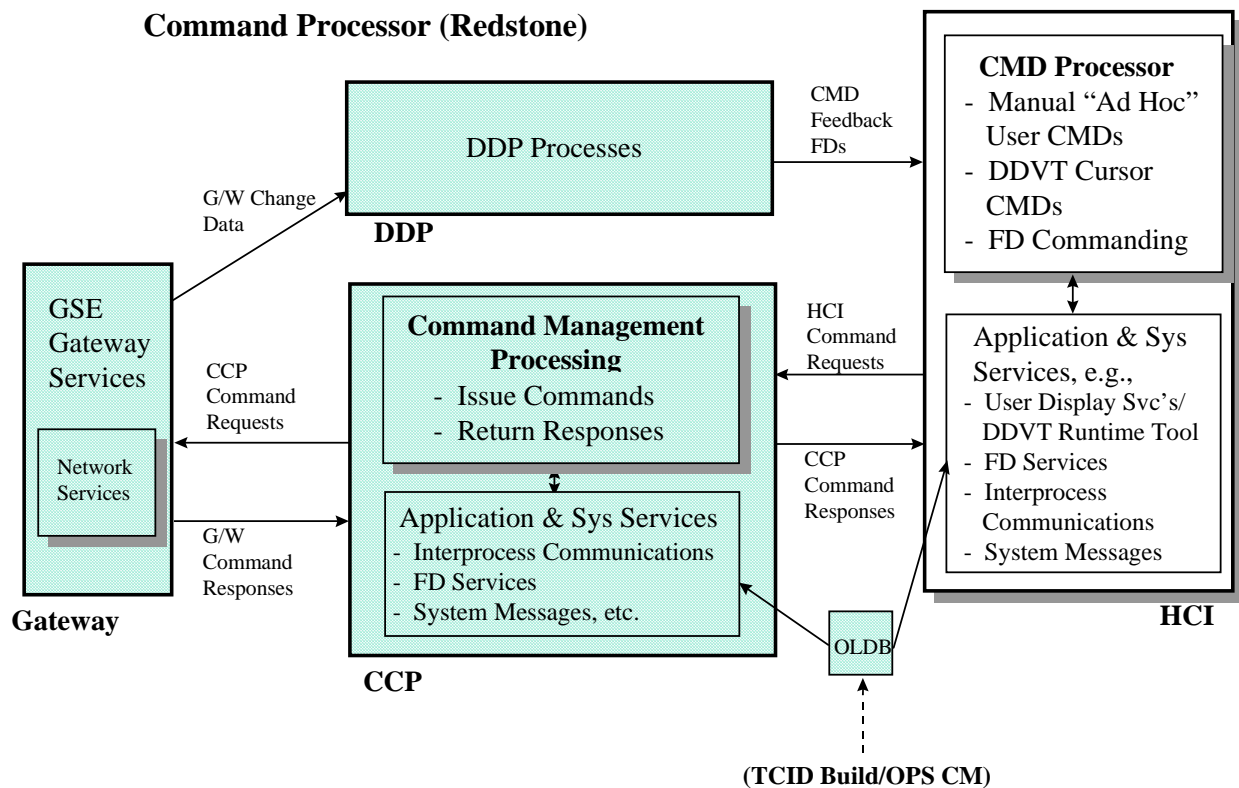
## 1. Command Processor

### 1.1 Command Processor Introduction

#### 1.1.1 Command Processor Overview

The Command Processor CSCI resides in the Human Computer Interface (HCI) and provides the capability for the HCI user to issue Command Requests to end items attached to the RTPS. The Command Processor CSCI receives CCP Command Responses for each Command Request, processes them and forwards the appropriate messages to User Display Services or System Message Services.

The data flow of the Command Processor is as follows:



#### 1.1.2 Command Processor Operational Description

The Command Processor provides a means to enter Command Requests to end items attached to the RTPS by an User at an HCI (workstation). The methods by which these Command Requests are entered will be prototype procedures for Redstone. Command Requests may be entered in two ways, manual entry and cursor clicks.

A Manual Entry Command Request is made by typing a text string into a field on a User display screen. The Command Processor CSCI parses the text string to create a Command Request. Error messages are displayed indicating syntax or command and FD accessibility errors. The Command Processor validates whether the Command Request is for an existing FD and is a supported command. The only FD Commands supported for Redstone are Set for Discrete FD's and Apply for Analog FD's. A Command Request will be sent if the Manual Entry is processed without error.

Cursor Click Command Requests are initiated by a cursor click of a "hot spot" or widget that represents a particular FD on a user display screen. Additional information concerning the Command Request will be acquired from the Command Processor

# Software Requirements Specification

User as needed through Pop-Up menus and data-entry windows. The Command Processor will utilize FD Services, User Display Services, and the DDVT Runtime Tool for support in structuring each Command Request.

Command Requests are then processed by Two Step Command Processing. (This processing prompts the user, if necessary, an additional time to either send or cancel the Command Request.) Once the Command Request leaves Command Processing, it is considered an HCI Command Request.

The HCI receives status about Command Requests at several stages in the command route. These messages consist of Two Step Command Processing error/status messages and CCP Command Responses. These messages/responses are displayed on the HCI screen through System Message Services and System Message Writer.

The Command Processor CSCI will be able to be run in either Stand-alone Application (Debug) Test or Full Test String operations. The HCI Command Request sent from the HCI (Command Processor) to the CCP and CCP Responses from the CCP to the HCI are accomplished using System Services Interprocess Communications. System Services Interprocess Communications makes the target process location for the request/response transparent to the Command Processor CSCI.

## 1.2 Command Processor Specifications

### 1.2.1 Command Processor Ground Rules

- 1.2.2.1 Command Requests supported for Redstone will not include communications to the following:
  - A) Prerequisite Control Logic
  - B) Test Application Scripts
  - C) End Item Manager
- 1.2.2.2 User Display Services will be required to support the Command Processor to:
  - A) initialize and terminate a GSE Display.
  - B) support Command Request widgets on the GSE Display.
  - C) provide pop-up menus for command options.
  - D) provide a modal pop-up window for 2 step command options.
  - E) provide modal pop-up windows for discrete FD input.
  - F) provide modal pop-up windows for analog FD input.
  - G) upon click of a widget will generate an event/call back.
  - H) provide the user the ability to cancel the window.
  - I) provide the associated FD command.
- 1.2.2.3 FD Services will be required to support the Command Processor to:
  - A) search for a FD by FD ID
  - B) search for a FD by FD Name
  - C) return the following FD attributes:
    - a) FD ID
    - b) FD Name
    - c) Current value
    - d) FD type
    - e) Options (for Discrete only - Open/ Close, On/Off, etc)
    - f) Range (for Analog only - Min, Max, precision)
    - g) Routing code
    - h) Destination
- 1.2.2.4 System Services Interprocess Communications will be required to support the Command Processor to provide communications between the Command Processor CSCI and the Command Management CSC.
- 1.2.2.5 The Command Processor CSCI will output error messages to System Message Services.
- 1.2.2.6 The Command Processor CSCI consists of user interfaces and background modules. The Command Processor CSCI user interfaces will be initialized by System Services Display Services. The Command Processor CSCI background modules will be initialized and terminated by System Control Ops Configuration Manager.

# Software Requirements Specification

## 1.2.1 Command Processor Functional Requirements

1.2.2.1 The Command Processor shall provide the facility to enter two types of FD commands.. The commands are *Set* for discrete FD's and the *Apply* for analog FD's.

### 1.2.2.2 Cursor Click Commands

- A) The Command Processor shall provide the facility to structure an HCI Command Request when a Widget (graphic object or "hot-spot") is selected by cursor click.
- B) After selection of a Widget, the Command Processor shall support further definition of command intent through the use of pop-up menus and text-entry screens.

### 1.2.2.3 Manual Entry Commands

- A) The Command Processor shall provide the facility to structure and send manually typed Command Requests.
- B) The commanded objects shall be designated by FD names.
- C) The Command Processor Set command shall use the correct discrete definition for the FD. For example:
  - a) Open/Close
  - b) True/False
  - c) Wet/Dry
  - d) On/Off
- D) The Command Processor Apply command shall use numeric input for analog FD's.

### 1.2.2.4 FD Commanding (Command Processing)

- A) The Command Processor shall provide each Command Request with sender's logical identifier (source).
- B) All Command Requests shall be sent from the HCI to the CCP. Failure to issue a command will be reported according to the table below.

		Visual Action	Additional Messages
a	HCI Failure to issue Command Request	Pop up Message Window	Message to System Message Services

- C) The Command Processor shall receive Command Responses from the CCP concerning each Command Request. The responses shall be routed to the User's Screen and System Message Services according the table below.

	CCP Command Response	Visual Action	Additional Messages from DDVT input	Additional Messages from Manual Entry Commands
a	Authentication Problem	Pop up Message Window	No Message	Message to System Message Services
b	Command Time-out	Pop up Message Window	No Message	Message to System Message Services
c	Command issued	TBD	No Message	Message to System Message Services
d	Command completed	TBD	No Message	Message to System Message Services
e	Command Error (non-HIM error)	Pop up Message Window	Message to System Message Services	Message to System Message Services
e	Command Error (HIM error)	Pop up Message Window	No Message	Message to System Message Services

- D) The Command Processor shall restrict invalid Command Requests
  - a) The Command Processor will reject invalid FD names.
  - b) The Command Processor will reject values out-of-range for an intended FD.
  - c) The Command Processor will reject invalid value types for an intended FD.
- E) The Command Processor shall support two-step Command Requests (Arm first, then execute) for

# Software Requirements Specification

manual entry commands.

- F) The Command Requests structure shall include:
- a) Time Command Request was prepared.
  - b) Logical User Identification.
  - c) Routing Code
  - d) Request ID
  - e) Payload Type
  - f) Source
  - g) Destination
  - h) FD ID
  - i) FD Value

## 1.2.3 Command Processor Performance Requirements

Command Processor Performance Requirements are not defined for the Redstone Release.

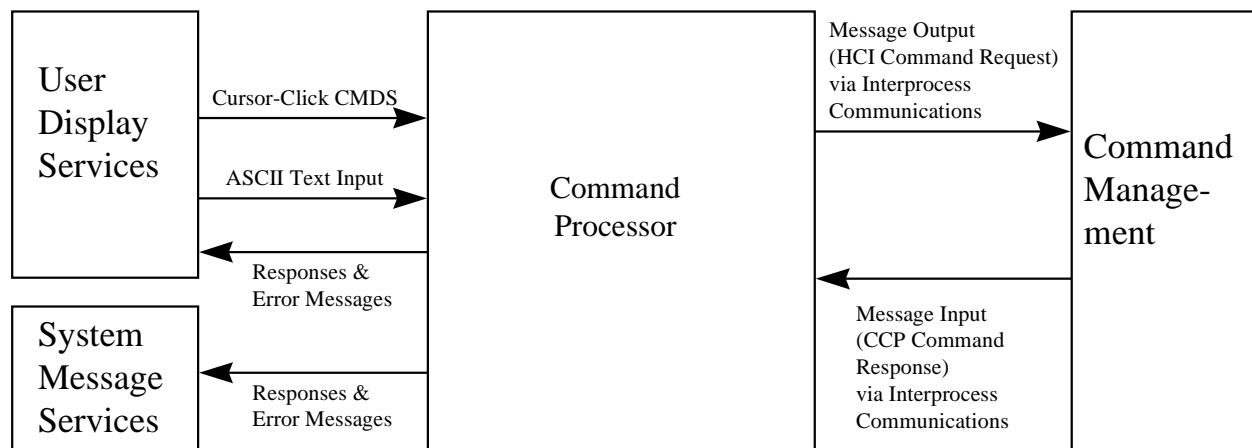
## 1.2.4 Command Processor Interfaces

Command Processor Interfaces are a set of API library function calls and structure as defined in the following documents:

System Services Interprocess Communications	MCC Event Services API Man Pages and Software User's Guide, LOR-TR1522, Volume I and II, Rev. 1.
FD Services	TBD
System Message Services	TBD

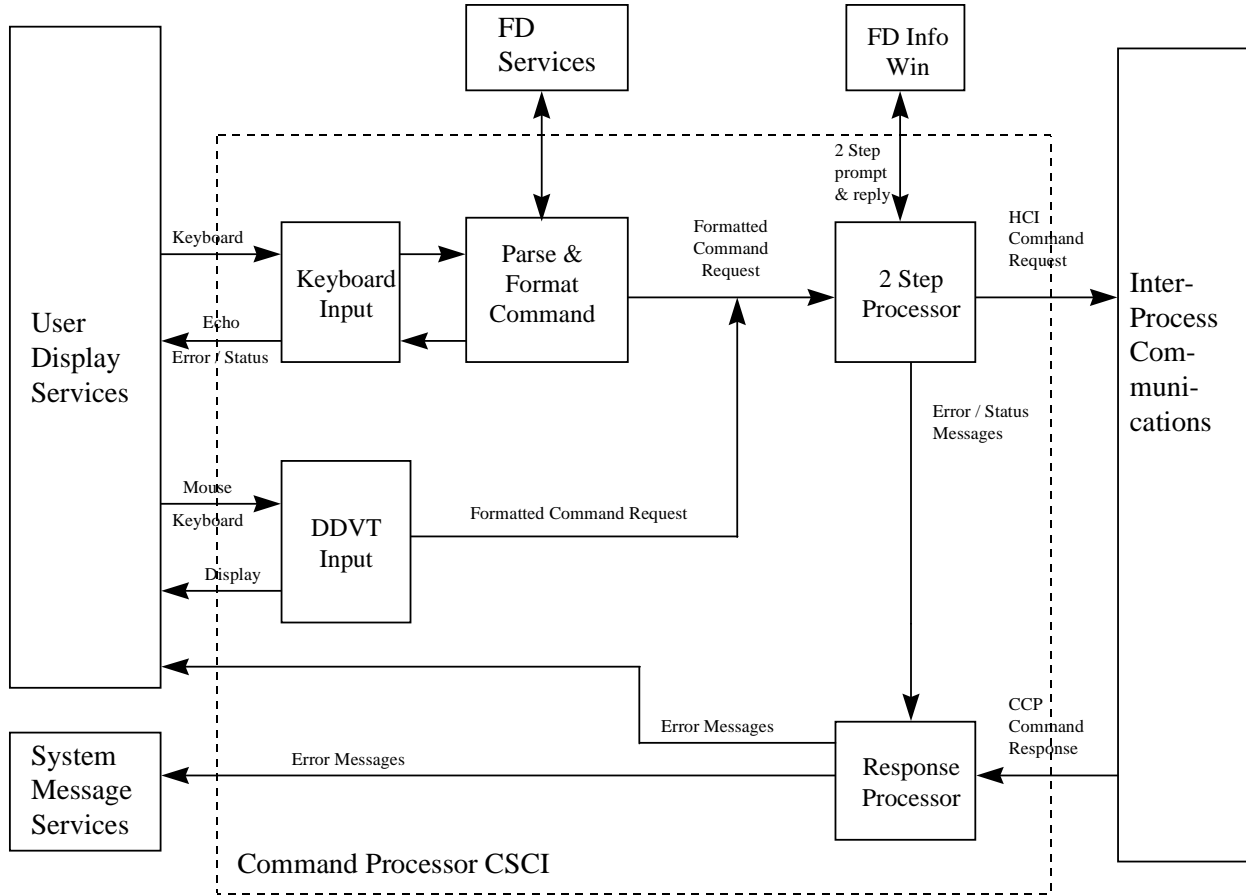
## 1.2.5 Command Processor Data Flow Diagram

### External



# Software Requirements Specification

## Internal



# Software Requirements Specification

## 1.2.6 Command Processor Structure Diagram

